



Netzbasierte Informationssysteme **Sprachen für Web-Anwendungen** **(XHTML, HTML5, XForms, CSS)**

Prof. Dr.-Ing. Robert Tolksdorf
Freie Universität Berlin
Institut für Informatik
Netzbasierte Informationssysteme
mailto: tolk@inf.fu-berlin.de
<http://www.robert-tolksdorf.de>





XHTML

Weiterentwicklung von HTML

- *XHTML* ist die Neuformulierung von HTML mit Hilfe einer XML-DTD (<http://www.w3.org/TR/xhtml1>)
- XHTML definiert einen XML Namensraum, in dem die bekannten Tags und Attribute von HTML 4 definiert sind
- Hauptsächlich syntaktische Änderungen wg. Anforderungen an Wohlgeformtheit
- Funktionale Ergänzungen des Sprachumfangs noch nicht
- Zukünftig: Modularisierung der Sprache des Web

XHTML...

- ... ist die Weiterentwicklung von HTML auf XML Basis
(Quelle: <http://www.w3.org/MarkUp/>)
- XHTML 1.0
 - Zweck: HTML 4 auf XML Basis
 - Status: W3C Recommendation 26 January 2000
 - Quelle: <http://www.w3.org/TR/xhtml1/>

- Modularization of XHTML
 - Zweck: Definition abstrakter Module von XHTML
 - Status: W3C Recommendation 10 April 2001
 - Quelle: <http://www.w3.org/TR/xhtml-modularization/>
- XHTML 1.1 - Module-based XHTML
 - Zweck: Definition konkreter Module von XHTML
 - Status: W3C Recommendation 31 May 2001
 - Quelle: <http://www.w3.org/TR/xhtml11/>

XHTML 1.0

- Neuformulierung von HTML 4, keine wesentlichen Änderungen der Tag-Menge
- XHTML Dokumente müssen im XML-Sinn wohlgeformt also korrekt geschachtelt sein
- Öffnenden Elemente müssen geschlossen werden (`<p>!`)
- Alle Element- und Attributnamen sind in Kleinschreibung
- Attributwerte müssen in Anführungszeichen stehen (z.B. `start="1"`)
- Attribute müssen immer einen Wert tragen (z.B. `checked="checked"`)
- Leere Tags entsprechen XML markiert: `
`
- `<scri pt>` und `<styl e>` enthalten #PCDATA, deshalb werden `<`, `&` und Kürzel (`&l t;`) interpretiert
- Fragmente verweisen immer auf das `i d`-Attribut

HTML4/XHTML1 Kompatibilität

- Leere Elemente durch Leerzeichen kompatibel machen:
`
` -> `
` -> `
`
- Abkürzung bei nichtleeren Elementen vermeiden:
`<p />` -> `<p><p />`
- `<script>` und `<style>` als CDATA markieren:
`<script>`
 `<![CDATA[Script]]>`
`</script>`
- Attribute `lang` und `xml:lang` gleichzeitig verwenden

Modularisiertes XHTML

- Structure Module
 - Geben Struktur eines Dokuments
 - `body`, `head`, `html`, `title`
- Text Module
 - Umfassen Elemente die Text enthalten
 - Überschriften: `h1`, `h2`, `h3`, `h4`, `h5`, `h6`
 - Blöcke: `address`, `blockquote`, `div`, `p`, `pre`
 - Laufender Text: `abbr`, `acronym`, `br`, `cite`, `code`, `dfn`, `em`, `kbd`, `q`, `samp`, `span`, `strong`, `var`
- Hypertext Module
 - Elemente für Hyperlinks
 - `a`

Modularisiertes XHTML

- List Module
 - Umfaßt Elemente zur Auszeichnung von Listen
 - dl , dt, dd, ol , ul , li
- Presentation Module
 - Elemente zur visuellen Markierung
 - b, i , tt, big, small , sub, sup, hr
- Edit Module
 - Elemente zur Markierung von Änderungen
 - del , ins
- Bi-directional Text Module
 - Schrift in zwei Richtungen
 - bdo

Modularisiertes XHTML

- Basic Forms Module
 - Einfache Formulare
 - `form`, `input`, `label`, `select`, `textarea`
- Forms Module
 - Formulare wie in HTML4
 - `form`, `input`, `label`, `select`, `textarea`, `fieldset`, `button`, `legend`, `optgroup`

Modularisiertes XHTML

- Basic Tables Module
 - Einfache Tabellen
 - `table`, `caption`, `td`, `th`, `tr`
- Tables Module
 - Tabellen wie in HTML4
 - `table`, `caption`, `td`, `th`, `tr`, `col`, `col group`, `tbody`, `thead`, `tfoot`

Modularisiertes XHTML

- Image Module
 - Abbildungen
 - `img`
- Client-side Image Map Module
 - Browser-seitige Image Maps
 - `a*`, `area`, `img*`, `input*`, `map`, `object*`
- Server-side Image Map Module
 - Server-seitige Image Maps
 - `img*`, `input*`,
- Object Module
 - Eingebettete Objekte
 - `object`, `param`

Modularisiertes XHTML

- Frames Module
 - Rahmen
 - frameset, frame, noframe
- Target Module
 - Rahmen als Ziele für Inhaltsdarstellung
 - a*, area*, base*, link*, form*
- Iframe Module
 - Eingebettete Rahmen
 - i frame

Modularisiertes XHTML

- Intrinsic Events Module
 - Ereignisse
 - `a*`, `area*`, `frameset*`, `form*`, `body*`, `label*`, `input*`, `select*`, `textarea*`, `button*`
- Metainformation Module
 - Metainformationen
 - `meta`
- Link Module
 - Metaverweise
 - `link`
- Base Module
 - Basis URL für Dokument
 - `base`

Modularisiertes XHTML

- Scripting Module
 - `noscript`, `script`
- Style Sheet Module
 - Formatvorlagen
 - `style` + `style` Attribute Module Änderungen

Modularisiertes XHTML

- Legacy Module
 - Überkommene Elemente
 - `basefont`, `font`, `center`, `menu`, `dir`, `index`, `s`, `strike`, `u`
 - `body*`, `br*`, `caption*`, `div*`, `dl*`, `h1-h6*`, `hr*`, `img*`, `input*`, `legend*`, `li*`, `ol*`, `p*`, `pre*`, `script*`, `table*`, `tr*`



- Führt keine neuen Elemente ein
- Umfasst alle Module aus XHTML Modularization
- Ist zweites Profil neben XHTML Basic
- Praktisch zunächst kaum Auswirkungen

- XHTML 2.0
 - Zweck: HTML 4 auf XML Basis
 - Status: W3C Working Draft 26 July 2006
 - Quelle: <http://www.w3.org/TR/xhtml2/>
- HTML 5
 - Zweck: HTML 4 evolutionär weiterentwickeln
 - Status: W3C Working Draft 15 November 2007
 - Quelle: <http://www.w3.org/html/wg/html5/>

- **W3C Launches New HTML Working Group**

2007-03-07: W3C is pleased to invite participation in the [new HTML Working Group](#), chartered to create the next HTML standard with the active participation of browser vendors, software developers, and content designers. "It's time to revisit the standard and see what we can do to [meet the current community needs, and to do so effectively with commitments from browser manufacturers in a visible and open way,](#)" said Tim Berners-Lee, W3C Director. [At the same time, W3C is chartering the Forms Working Group, the XHTML2 Working Group, and rechartering the Hypertext Coordination Group.](#) Read the press release and visit the [HTML Working Group home page](#), the [Forms Working Group home page](#), and the [XHTML2 Working Group home page](#).

[www.w3c.org]

- „A particular case is HTML. HTML has the potential interest of millions of people: anyone who has designed a web page may have useful views on new HTML features. It is the earliest spec of W3C, a battleground of the browser wars, and now the most widespread spec. [...]
- Some things are very clear. It is really important to have real developers on the ground involved with the development of HTML. It is also really important to have browser makers intimately involved and committed. And also all the other stakeholders, including users and user companies and makers of related products.
- Some things are clearer with hindsight of several years. **It is necessary to evolve HTML incrementally. The attempt to get the world to switch to XML, including quotes around attribute values and slashes in empty tags and namespaces all at once didn't work.** The large HTML-generating public did not move, largely because the browsers didn't complain. Some large communities did shift and are enjoying the fruits of well-formed systems, but not all. It is important to maintain HTML incrementally, as well as continuing a transition to well-formed world, and developing more power in that world.
- The plan is to charter a completely new HTML group. Unlike the previous one, this one will be chartered to do **incremental improvements to HTML**, as also in parallel xHTML. It will have a different chair and staff contact. It will work on HTML and xHTML together. We have strong support for this group, from many people we have talked to, including browser makers.“

- Designziele
 - XML Konzepte besser nutzen
 - Stylesheets zur Darstellung, XHTML2 für Struktur
 - Praktikablere Syntax
 - Barrierefreiheit, Internationalisierung
 - Geräteunabhängigkeit
 - Deklarative Alternativen zu Scripten
 - Integration mit Semantic Web

- Strukturierungsmöglichkeiten:
 - `<section>` Element für Abschnitte
 - `<h>` Element für Überschrift
 - `<l>` Element für umbrechbare Zeilen
 - `<p>` Element auch für z.B. Listen oder Tabellen
- `<nl>` Element für Navigationsleisten
- Alle Elemente können Quellanker sein
- Metadaten: role Attribut, meta und link nutzen RDF
- Formulare -> XForms
- Framesets -> XFrames
- ...

- Working Draft lässt noch vieles offen

- Entstanden aus Analyse existierender Browser und Inhalte
- Eigenschaften:
 - HTML 5 sowohl in „HTML-Syntax“ als auch als XML
 - Klare Verarbeitungsmodelle
 - Verbesserung der Auszeichnungsmöglichkeiten
 - DOM API Anpassungen
(HTMLDocument Interface in DOM Level 2 HTML)
- [HTML 5 differences from HTML 4. Editor's draft 22 October 2007. <http://www.w3.org/html/wg/html5/diff/>]

Syntax

- HTML als text/html

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Example document</title>
  </head>
  <body>
    <p>Example paragraph</p>
  </body>
</html>
```
- XML als application/xml oder application/xhtml+xml

```
<?xml version="1.0" encoding="UTF-8"?>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Example document</title>
  </head>
  <body>
    <p>Example paragraph</p>
  </body>
</html>
```

Neue Elemente / Struktur

- `<section>`: Dokumentenabschnitt (siehe `<h1>...<h6>`)
- `<header>`: Überschrift eines Abschnitts
- `<footer>`: „Unterschrift“ eines Abschnitts
- `<article>`: Artikelähnlicher Dokumentenabschnitt (-> blogs)

- `<aside>`: „Infokasten“ im Dokument
- `<nav>`: Navigationsabschnitt einer Seite
- `<figure>`: Abbildung als Element mit Beschriftung

- `<dialog>`: Konversation als Liste

Neue Elemente / Inhalte

- `<audio>` und `<video>`: Multimediale Inhalte mit Scripting-Schnittstelle
- `<embed>`: Eingebettete Objekte
- `<meter>`: Anzeigeskala
- `<time>`: Zeit oder Datum
- `<canvas>`: Zeichenfläche
- `<command>`: Aufrufbares Kommando
- `<datagrid>`: Interaktive Repräsentation von Daten
- ...

Neue Eingabefelder

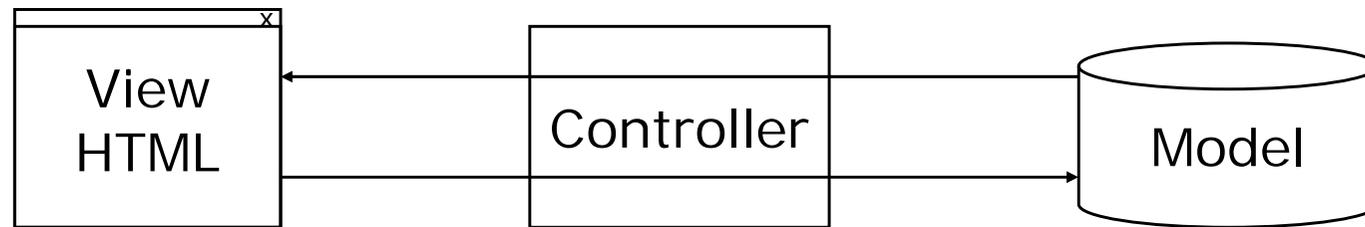
- Das type Attribut bei `<input>` kann neue Werte tragen:
datetime
 - datetime-local
 - date
 - month
 - week
 - time
 - number
 - range
 - email
 - url



XForms

- XForms 1.0
 - Zweck: Formularbasierte Interaktion zwischen menschlichem Nutzer und Agenten im Netz
 - Status: W3C Recommendation 29 October 2007
 - Quelle: <http://www.w3.org/TR/xforms/>

- Model-View-Controller Muster im Web:



- In HTML können die Interaktionen nur schlecht an die Typisierung des Modells gebunden werden
 - Untypisierte Eingabefelder
 - Vermischung von Aspekten (Layout, Position und Inhalt von Formularen)
 - Keine Anbindung zur „Datenwelt“ von XML

XForms

- Trennung von
 - Einzugebenen Daten
 - Layout des Formulars und der Eingabefelder
- Daten sind unabhängig vom Formular
 - Können auf standardisierte Weise übermittelt werden
 - Können wiederverwendet werden
- XForms ist eigenständige XML-Sprache
 - In XHTML nutzbar
 - In anderen XML-Sprachen nutzbar
- Vorteile
 - Starke Typisierung im Client prüfbar
 - Schema Wiederverwendung
 - Barrierefreiheit, Internationalisierung
 - Geräteunabhängigkeit
 - Deklarative Alternativen zu Scripten

- Beispiel [<http://www.w3.org/TR/xforms/>]

Select Payment Method: Cash Credit

Credit Card Number:

Expiration Date:

- Das Formular beinhaltet die Informationen
 - Zahlungsweise
 - Kreditkartennummer
 - Ablaufdatum der Karte
 - Verweise auf Verarbeitung bei Abschicken durch Submit

- Formularmodell in XForms:

```
<xforms:model>
```

```
<xforms:instance>
```

```
<ecommerce xmlns="">
```

```
<method/>
```

```
<number/>
```

```
<expiry/>
```

```
</ecommerce>
```

```
</xforms:instance>
```

```
<xforms:submission action="http://example.com/submit"  
  method="post" id="submit"  
  includenamespaceprefixes=""/>
```

```
</xforms:model>
```

HTML Formulare nach XHTML

- [Steven Pemberton. XForms for HTML Authors, Part 1/2.
<http://www.w3.org/MarkUp/Forms/2006/xforms-for-html-authors>,
<http://www.w3.org/MarkUp/Forms/2006/xforms-for-html-authors-part2>]

```
<html>
<head><title>Search</title></head>
<body>
  <form
    action="http://example.com/search"
    method="get">
    Find
  <input type="text" name="q">
    <input type="submit" value="Go">
  </form>
</body>
</html>
```

```
<h:html xmlns:h="http://www.w3.org/1999/xhtml"
  xmlns="http://www.w3.org/2002/xforms">
<h:head>
  <h:title>Search</h:title>
  <model>
    <submission action="http://example.com/search"
      method="get" id="s"/>
  </model>
</h:head>
<h:body>
  <h:p>
    <input ref="q"><label>Find</label></input>
  <submit
    submission="s"><label>Go</label></submit>
  </h:p>
</h:body>
</h:html>
```

HTML Formulare nach XHTML

- Eingabefelder:
 - First name: `<input type="text" name="firstname" >`
 - `<input ref="firstname" > <label>First name: </label> </input >`
- Auswahlfelder:
 - Gender:
`<input type="radio" name="sex" value="M" > Male`
`<input type="radio" name="sex" value="F" > Female`
 - `<select1 ref="sex" >`
 `<label >Gender: </label >`
 `<item >`
 `<label >Male </label > <value >M </value >`
 `</item >`
 `<item >`
 `<label >Female </label > <value >F </value >`
 `</item >`
 `</select1 >`
- ...

- Mehrfache Verarbeitungsmöglichkeiten

```
<model>
```

```
<instance><data xmlns=""><q/></data></instance>
```

```
<submission action="http://example.com/search"  
  method="get" id="com"/>
```

```
<submission action="http://example.org/search"  
  method="get" id="org"/>
```

```
</model>
```

```
<submit submission="org"><label>Search  
  example.org</label></submit>
```

```
<submit submission="com"><label>Search  
  example.com</label></submit>
```

Starke Typisierung

- Wertebereich einschränken binden:

```
<model>
```

```
  <instance> <data xmlns="">
```

```
    <year/>
```

```
  </data> </instance>
```

```
  <bind nodeset="cc" relevant="../method='credit'"/>
```

```
  <bind nodeset="expires" relevant="../method='credit'"/>
```

```
</model>
```

```
  <bind nodeset="year" constraint=". > 1970"/>
```

- Errechnete Werte:

```
  <bind ref="volume"
```

```
    calculate="../height * ../width * ../depth"/>
```

- Typisierung von Felder:

```
  <bind nodeset="q" type="xsd:integer"/>
```

- Eigene XML-Schema Datentypen verwendbar

```
<model>
```

```
  <instance> <data xmlns=""> <flw/> </data> </instance>
```

```
  <schema xmlns="http://www.w3.org/2001/XMLSchema">
```

```
    <simpleType name="curse">
```

```
      <restriction base="xsd:string">
```

```
        <pattern value="[a-z][a-z][a-z][a-z]"/>
```

```
      </restriction>
```

```
    </simpleType>
```

```
  </schema>
```

```
  <bind nodeset="flw" type="curse" />
```

```
  ...
```

```
</model>
```

Alternativfelder ohne Scripting

```
<switch>
  <case id="start" >
    <group>
      <label>About you</label>
      <input ref="name" > <label>Name: </label> </input>
    [...]
    </group>
    <trigger>
      <label>Next</label>
      <toggle case="preferences" ev:event="DOMActivate"/>
    </trigger>
  </case>
  <case id="preferences">
    <group>
      <label>Your preferences</label>
      <input ref="food" > <label>Food: </label> </input>
    [...]
    </group>
    <trigger>
      <label>Next</label>
      <toggle case="history" ev:event="DOMActivate"/>
    </trigger>
  </case>
[...]
```

About you

Name:

City:

Email:

Your preferences

Food:

Drink:

Music:



Darstellung von Inhalten

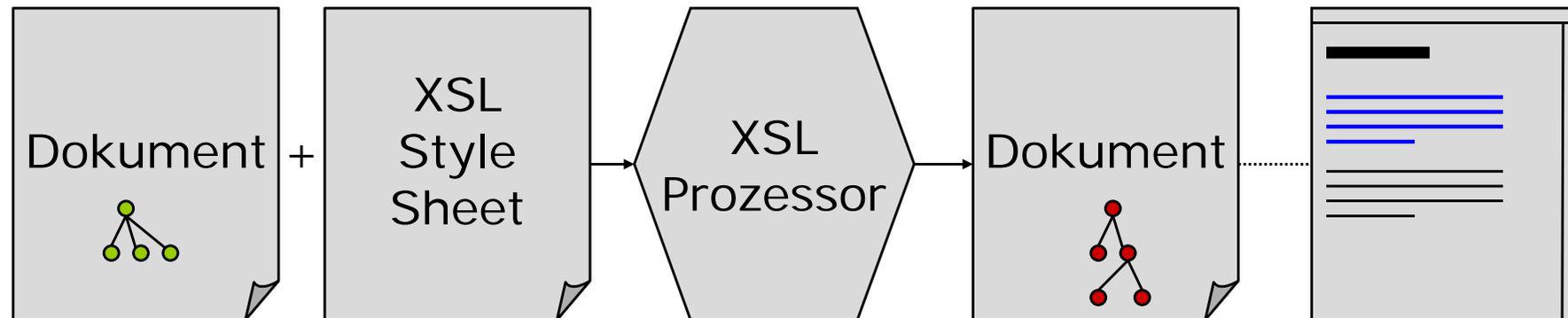
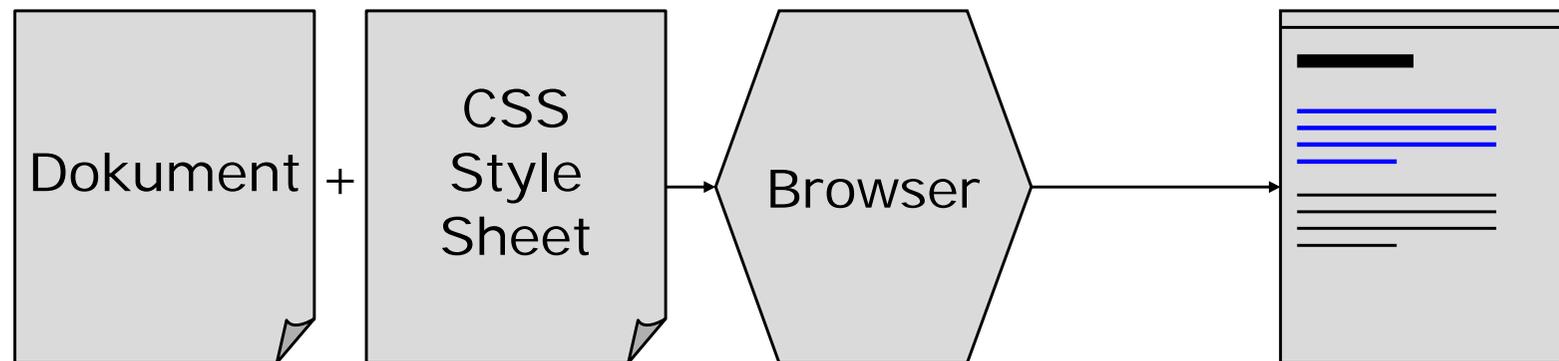
Inhalt und Darstellung

- Auszeichnungssprachen markieren
 - logische Struktur (<h1>, <p>, <table>)
 - Darstellung (, <i>, , <big>)
- Beispielhaftes Problem:
...
 - Keine Auswirkung beispielsweise mit Lynx
 - nicht lesbar bei kleinem Display
 - Suchmaschinen erkennen „Betonung“ nicht
 - nicht mit allen Seitenhintergrundfarben lesbar
 - wählt nicht vorhandenen oder nicht passenden Font aus

- Trennung von Inhalt und Darstellung ist aber wichtig für
 - Geräteunabhängigkeit von Informationen (Handy vs. PC)
 - Medienunabhängigkeit von Informationen (Grafik vs. Sprache)
 - Sprachunabhängigkeit von Informationen (" vs. „ vs. »)
 - Mehrkanal Veröffentlichungen (WAP und Web)
 - Verarbeitbarkeit von Informationen
- Notwendig:
 - Darstellungsmodelle
 - Sprachen zur Festlegung von Darstellungseigenschaften
 - Verarbeitungsmodelle für diese Sprachen

Style Sheets

- W3C Style Activity entwickelt
 - Cascading Stylesheets CSS
 - XML Stylesheet Language XSL





Cascading Style Sheets CSS

Cascading Style Sheets CSS

- ... Mechanismus zur separaten Definition von Stileigenschaften für HTML und XML Dateien (Quelle: <http://www.w3.org/Style/CSS/>)
- Cascading Style Sheets, level 1
 - Ziel: Sprache zur Definition des Darstellungsstils von HTML Dokumenten
 - Status: W3C Recommendation 17 Dec 1996, revised 11 Jan 1999
 - Quelle: <http://www.w3.org/TR/REC-CSS1>
- Cascading Style Sheets, level 2
 - Ziel: Sprache zur Definition des Darstellungsstils von HTML und XML Dokumenten für unterschiedliche Medienarten
 - Status: W3C Recommendation 12-May-1998
 - Quelle: <http://www.w3.org/TR/REC-CSS2>
- Cascading Style Sheets, level 3
 - Ziel: Modularisierte und erweiterte Sprache zur Definition des Darstellungsstils von HTML und XML Dokumenten
 - Status: „Under Construction“, unterschiedlicher Entwicklungsstand
 - Quelle: <http://www.w3.org/Style/CSS/current-work>

CSS Grundidee

- Grundidee:
Zu HTML Tags werden definierte Attribute für Darstellungseigenschaften gesetzt
- CSS-Datei getrennt von HTML-Datei gehalten
- Beispiel: Um Überschriften in großer blauer Schrift darzustellen:

```
h1 {color: blue; font-size: 22pt; }
```
- CSS definiert
 - Rahmensyntax zur Notation
 - Menge von Attributen
 - Menge von Werten
 - Bedeutung
 - Mechanismen zur Anbindung von Stilinformationen an und in HTML Seiten

CSS Anbindung an HTML

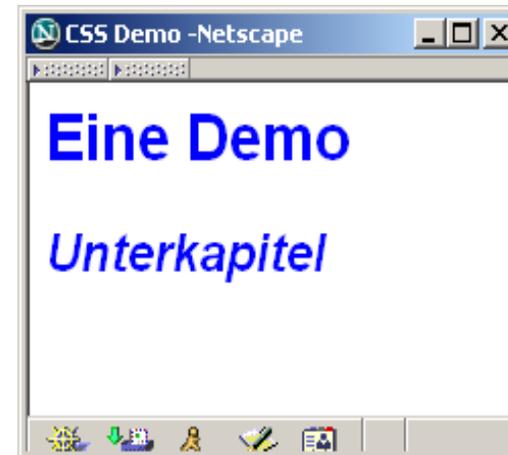
- Drei Wege der Einbindung in HTML
 - Mit externem Stylesheet über Verweis im `<link>`-Tag:
`<link rel="stylesheet" type="text/css" href="http://www.ag-nbi.de/incl ude/nbi .css">`
 - Im HTML Dokument mit dem `<style>`-Tag:
`<style>`
`h1 {color: blue; font-size: 22pt; }`
`</style>`
Kompatibel für alte Klienten:
`<style><!--`
`h1 {color: blue; font-size: 22pt; }`
`--></style>`
 - Bei den einzelnen Elementen im `style`-Attribut:
`<h1 style="color: red">Rote Überschrift</h1>`
- Einbindung innerhalb eines CSS
 - `@import url (http://www.ag-nbi.de/incl ude/col ors. css);`

CSS Gruppierungen und Vererbung

- In CSS können Angaben gruppiert werden
 - Mehrere Elemente erhalten gleiche Eigenschaften
`h1, h2, h3, h4, h5, h6 {color: blue; }`
 - Ein Element erhält mehrere Eigenschaften
`h1 {color: blue; font-size: 22pt; }`
 - Kombination
`h1, h2, h3, h4, h5, h6 {color: blue; font-style: italic}`
- Eigenschaften werden „vererbt“
 - Entlang der Element Schachtelung nach „unten“
`<h1>Das ist <u>wichtig</u></h1>`
 - Allgemeinstes Element in HTML: `<body>`
 - Eigenschaften die dort gesetzt werden, gelten für alle Elemente bei denen nichts anderes deklariert ist
 - Beispiel: Seite komplett in serifenloser Schrift
`body {font-family: arial, helvetica, sans-serif; }`

- HTML:

```
<html >
<head>
  <link rel="stylesheet" type="text/css" href=
    "cssdemo.css">
  <title>CSS Demo</title>
</head>
<body>
  <h1>Eine Demo</h1>
  <h2>Unterkapitel </h2>
</body>
```



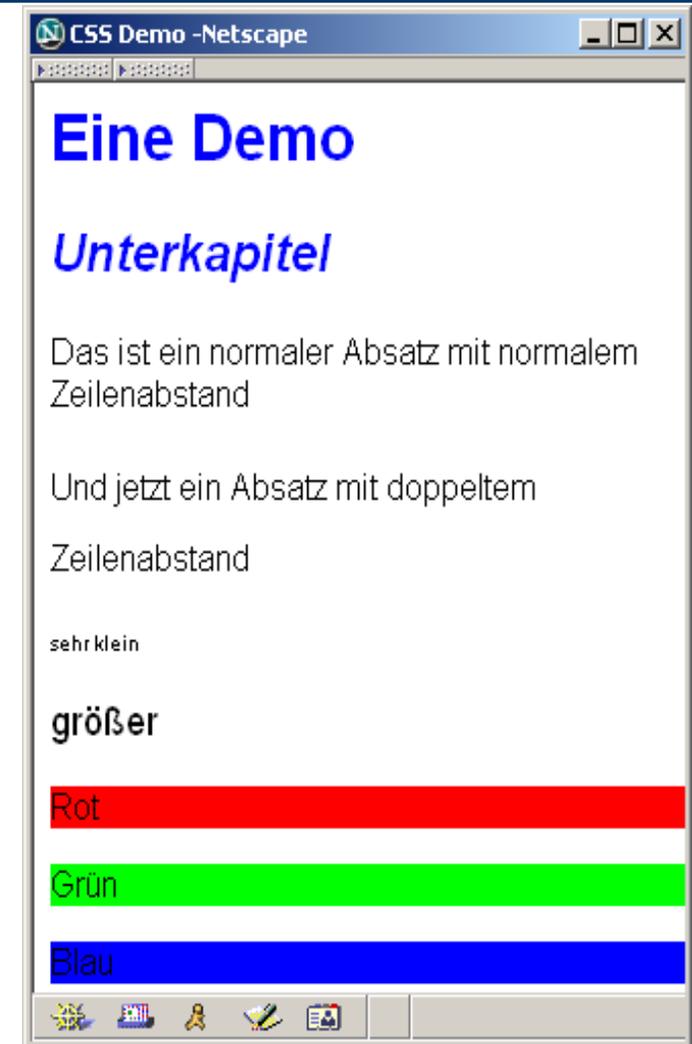
- CSS:

```
body {font-family: arial, helvetica, sans-serif; }
h1 {color: blue; font-size: 22pt; }
h2, h3, h4, h5, h6 {color: blue; font-style: italic}
body {font-family: arial, helvetica, sans-serif; }
```

Wertetypen

- Längen ([+|-]<Zahl><Einheit>)
 - Relative Längen
 - em: Breite des M im aktuellen Font
 - ex: Höhe des x im aktuellen Font
 - px: Referenzpixel auf einem 90 dpi Gerät
 - Absolute Längen
 - in: Zoll (1in=2,54cm)
 - cm: Zentimeter
 - mm: Millimeter (10mm=1cm)
 - pt: Typographischer Punkt (1pt=1/72in)
 - pc: Pica (12pc=1pt)
- Anteilige Größen in Prozent
 - Bei Eigenschaft ist Bezug definiert
 - p { line-height: 120% }: Relativ zum aktuellen font-size
 - h1 { margin-right: 12.3% }: Relativ zum aktuellen margin-right


```
<p>Das ist ein normaler Absatz mit normaler Zeilenabstand</p>  
<p style="line-height: 200%">Und jetzt ein Absatz mit doppeltem Zeilenabstand</p>  
<p style="font-size: x-small">sehr klein</p>  
<p style="font-size: larger">größer</p>  
<p style="background-color: red">Rot</p>  
<p style="background-color: #0F0">Grün</p>  
<p style="background-color: rgb(0, 0, 100%)">Blau</p>
```



CSS Klassen

- Darstellungseigenschaften für alle Verwendungen eines bestimmten Elements gleich
- Tatsächlich aber vielleicht unterschiedliche Eigenschaften je nach Verwendung → Klassen in CSS
- Deklaration durch Punkt getrennt hinter Element:
`h1.largegreen {color: green; font-size: 22pt; }`
- Verwendung durch class-Attribut
`<h1 class="largegreen">Das ist jetzt anders</h1>`
- Deklaration von Eigenschaftsklassen ohne Elementangabe:
`.green {color: green; }`
- Verwendung bei allen Elementen möglich:
`<p class="green">Alles klar</p>`

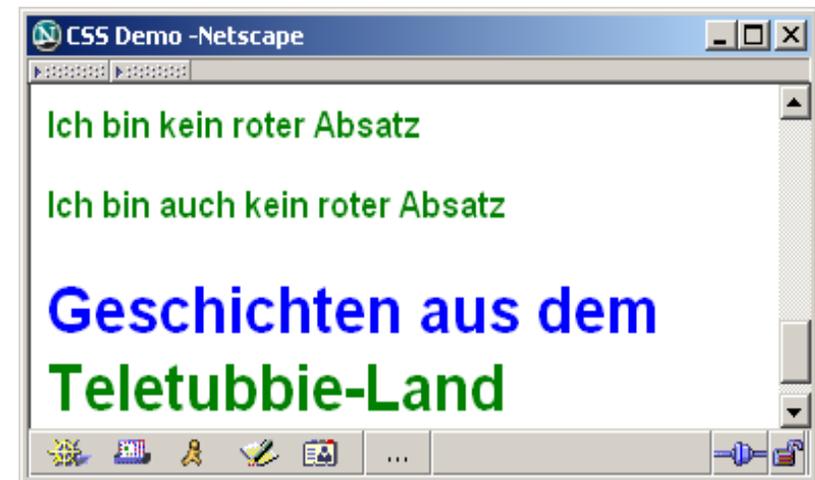
- Zwei Elemente in HTML, mit denen Stil gebunden werden kann ohne dass Darstellung erzeugt wird:
 - `<div>`: Generisches Block-Element, das andere Blöcke umschließt
 - ``: Generisches Inline-Element, das im fortlaufenden Text verwendet werden kann
- Attribute: `id`, `lang`, `dir`, `title`, `align`, `onclick`, `ondblclick`, `onmousedown`, `onmouseup`, `onmouseover`, `onmousemove`, `onmouseout`, `onkeypress`, `onkeydown`, `onkeyup`, `style`, `class`
- `<div class="green">`
`<p>bl ah bl ah</p><p>foo foo</p>`
`</div>`
- `<p>Das ist wirklich sehr`
`wichti g. </p>`

Demo

```
<p class="green"><b>I ch  
bi n kei n roter Absatz</b></p>
```

```
<di v class="green" >  
<p><b>I ch bi n auch kei n  
roter Absatz</b></p>  
</di v>
```

```
<h1>Geschi chten aus dem  
<span class="green">Tel etubbi e-Land</span></h1>
```



CSS Ausnahmen

- Unterschiedliche Eigenschaften je nach Kontext der Verwendung → „contextual selectors“ in CSS
- In Überschrift
`<h1>Das ist wichtig</h1>`
soll Hervorhebung farblich sein:
`h1 em { color: red }`
- Kann verknüpft werden
`ul li { font-size: small }`
`ul ul li { font-size: x-small }`
- Kann auch gemischt werden
`.red h1 {color: blue}`
→ `<h1>` in einem `<div class="red">` ist trotzdem blau

- Pseudoklassen (hier beim Anker-Element <a>)
 - a: l i n k: noch nicht besuchter Zielanker
 - a: v i s i t e d: schon besuchter Zielanker
 - a: a c t i v e: gerade ladenender Zielanker
 - a: h o v e r: Mauszeiger ist über Element
 - a: f o c u s: Element hat Eingabefocus
 - a: l i n k { t e x t - d e c o r a t i o n : n o n e ; }
 - a: h o v e r { t e x t - d e c o r a t i o n : u n d e r l i n e ; }
- Typographische Pseudoelemente
 - f i r s t - l e t t e r Erster Buchstabe
 - <p>Guten Tag</p> wird konzeptionell expandiert zu <p><p: f i r s t l e t t e r>G</p: f i r s t l e t t e r>uten Tag</p>
 - f i r s t - l i n e Erste Zeile
 - p: f i r s t - l i n e { f o n t - f a m i l y: s e r i f }
 - The first line of an article in Newsweek.
 - f i r s t - c h i l d Erstes Unterelement
- Können mit eigenen Klassen kombiniert werden:
p: e i n l e i t u n g: f i r s t - l i n e { f o n t - f a m i l y: s e r i f }

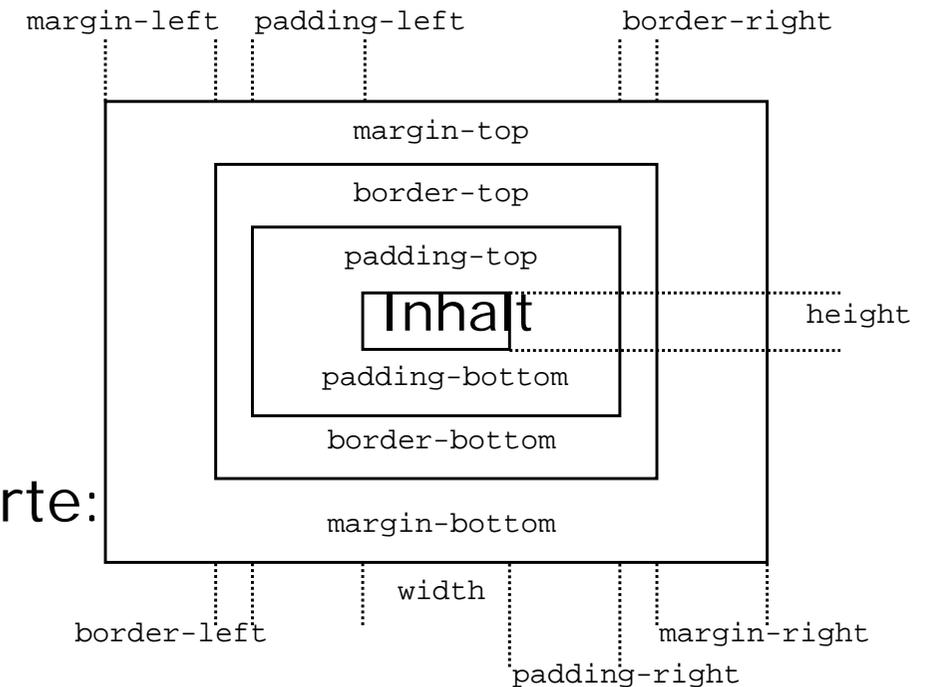
Auflösung von Konflikten

- Darstellungseigenschaften werden an verschiedenen Stellen definiert (Browser, Nutzerpräferenzen, Autor)
- Bei Konflikten gilt folgende ansteigende Gewichtung:
 - Defaults des Darstellers
 - Eventuell persönliches Stylesheet
 - Stylesheet des Dokuments
- Innerhalb von Stylesheets
 - Deklarationen mit Markierung !important wichtiger als solche ohne
(h1 { color: blue ! important ; font-size: 22pt; })
 - Deklarationen aus höher gewichteter Quelle sind wichtiger als solche aus anderen
 - Spezifischere Deklarationen gewichtiger als allgemeinere
(h1 em { color: red } vs. h1 { color: blue; })
 - Textuell spätere Deklaration gewichtiger als frühere (dabei @import am Anfang in ihrer Reihenfolge berücksichtigt)

Eigenschaften von Inhaltskästen

- Leerraum um Kasten

- margin-top: Oben
- margin-right: Rechts
- margin-bottom: Unten
- margin-left: Links
- margin: Zusammengefasste Werte:
 - margin: 1cm
Auf allen Seiten 1cm
 - margin: 1cm 2cm
Oben und unten 1cm, links und rechts 2cm
 - margin: 1cm 2cm 3cm
Oben 1cm, rechts 2cm, unten 3cm, links 2cm
 - margin: 1cm 2cm 3cm 4cm
Oben 1cm, rechts 2cm, unten 3cm, links 4cm



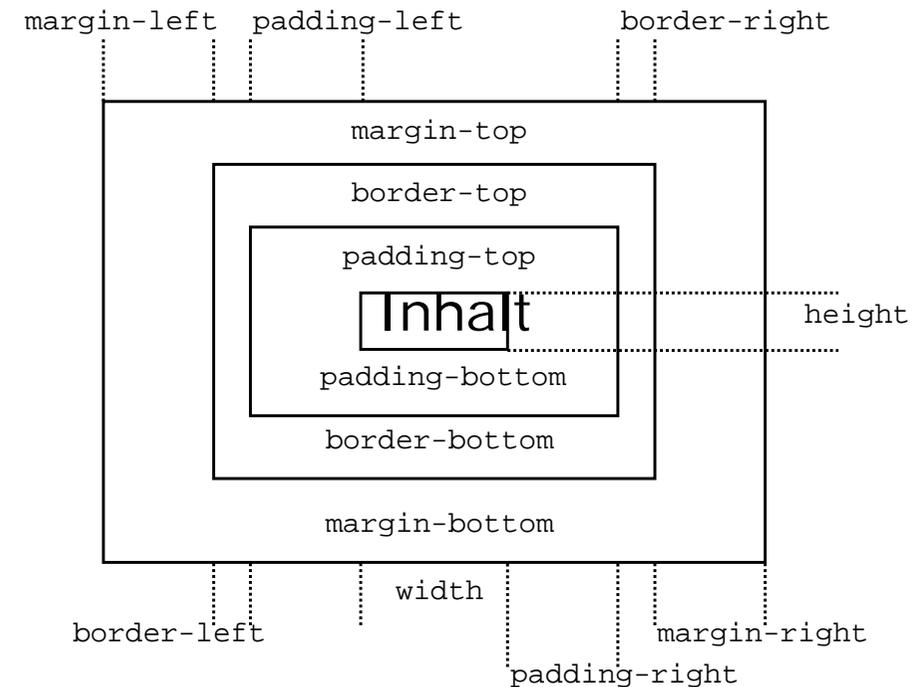
Eigenschaften von Inhaltskästen

- Leerraum um Inhalt

- padding-top: Oben
- padding-right: Rechts
- padding-bottom: Unten
- padding-left: Links
- padding:

Zusammengefasste Werte:

- padding: 1cm
Auf allen Seiten 1cm
- padding: 1cm 2cm
Oben und unten 1cm, links und rechts 2cm
- padding: 1cm 2cm 3cm
Oben 1cm, rechts 2cm, unten 3cm, links 2cm
- padding: 1cm 2cm 3cm 4cm
Oben 1cm, rechts 2cm, unten 3cm, links 4cm



Eigenschaften von Inhaltskästen

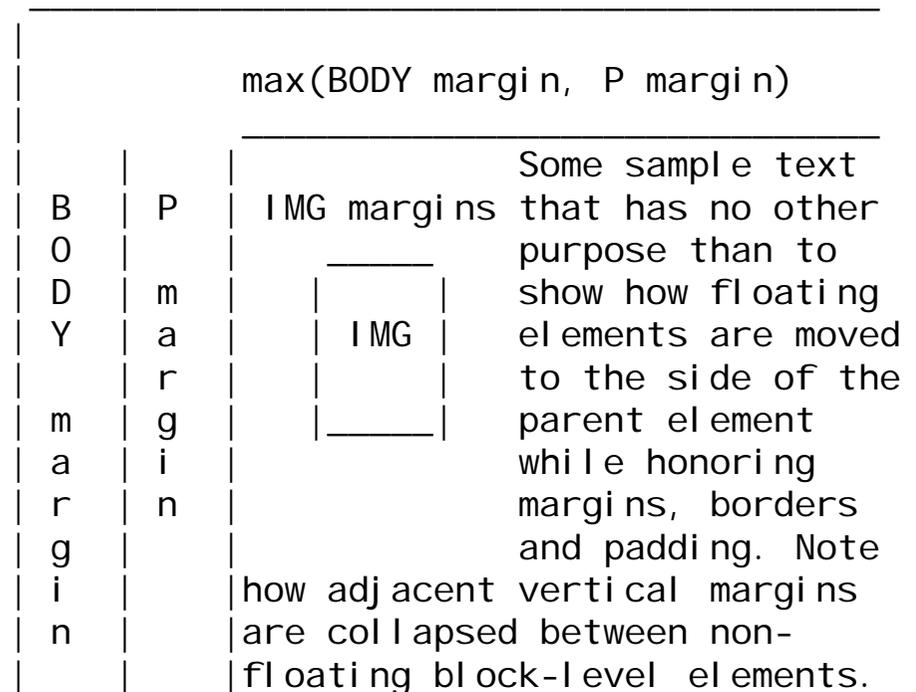
- Dicke des Rands um Inhalt + Padding
 - border-top-wi dth: *Dicke Oben*
 - border-ri ght-wi dth: *Dicke Rechts*
 - border-bott om-wi dth: *Dicke Unten*
 - border-l eft-wi dth: *Dicke Links*
 - border-wi dth: *oben rechts unten links* Zusammengefasst
 - Werte: thi n, medi um, thi ck oder Länge
- Farbe des Rands
 - border-col or: *oben rechts unten links*
- Linenstil des Rands
 - border-styl e: *oben rechts unten links*
 - none, dotted, dashed, solid, doubl e, groove, ri dge, inset, outset
- Zusammengefasst für Seiten einzeln
 - border-top: *Dicke Linienstil Farbe*
 - Gleiches für border-ri ght, border-bott om, border-l eft
 - border: *Dicke Linienstil Farbe* für alle vier Seiten gleich

Eigenschaften von Inhaltskästen

- `width`: *Länge* Breite des Inhalts
- `height`: *Länge* Höhe des Inhalts
- `float`: *Position* Inhalt kann an den Rand des enthaltenden Elements verschoben werden

```
<style type="text/css">
  img { float: left }
  body, p, img { margin: 2em }
</style>
<body>
  <p>
  Some sample text that has no
  other...</p></body>
```

- Werte: `left`, `right`, `none`
- `clear`: *Position* Ob an den Rand dieses Element Inhalt verschoben werden kann
 - Werte `none`, `left`, `right`, `both`



- CSS:

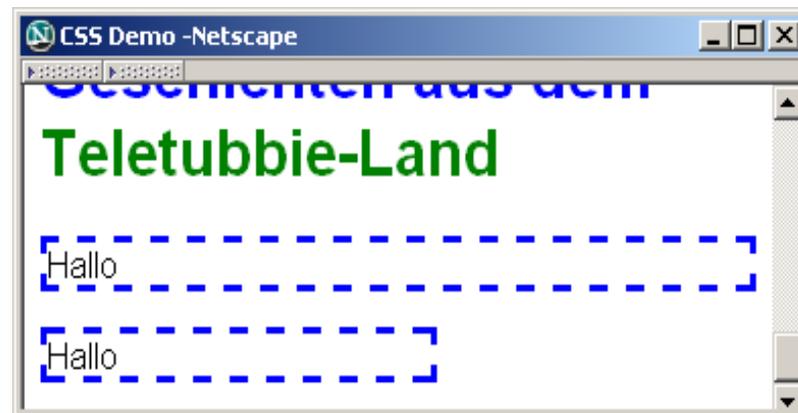
```
.aufwaendig {border: medium dashed blue; }
```

- HTML:

```
<p class="aufwaendig">Hallo</p>
```

```
<p class="aufwaendig"
```

```
style="width: 5cm; height: 2cm">Hallo</p>
```



Darstellungsklassen und Listen

- `display` Legt Darstellungsklasse eines Elements fest
 - `block`: Erzeugt Darstellungsblock (`<p>...</p>`)
 - `inline`: Wird innerhalb eines Blocks dargestellt (`<i>...</i>`)
 - `list-item`: Ist Listeneintrag (`...`)
 - `none`: Keines der obigen (``)
- Falls `display`: `list-item` gilt
 - `list-style-type` Nummerierungsart
 - Werte `disc`, `circle`, `square`, `decimal`, `lower-roman`, `upper-roman`, `lower-alpha`, `upper-alpha`, `none`
 - `list-style-image`: *URL* Bild als Markierung
 - `list-style-position` Markierungsposition relativ zum Block
 - Wert: `inside`

- Blah
 - foo blah
 - blah

outside

- Blah
 - foo
 - blah
 - `list-style` *Nummerierungsart URL Markierungsposition*
 - Kurznotation

Elementfarbe und Hintergrund

- `color`: *Farbe* Farbe des Elements
- `background-color`: *Farbe* Hintergrundfarbe
- `background-image`: *URL* Hintergrundbild
- `background-repeat` Wird das Hintergrundbild wiederholt und wie?
 - Werte: `repeat`, `repeat-x`, `repeat-y`, `no-repeat`
- `background-attachment` Ist Hintergrundbild verankert oder wird es mitgescrollt
 - Werte: `scroll`, `fixed`
- `background-position`
 - Werte: Absolute Länge oder relativ zur Umgebungsgröße
 - Schlüsselworte aus
 - `top`, `center`, `bottom`
 - `left`, `center`, `right`
 - `center right` entspricht 100% 50%

Elementfarbe und Hintergrund

- background *Farbe Wiederholung Fixierung Position*
 - Kurznotation
- Institutshomepage:

```
body {  
  color : #222222;  
  background : #ffffff  
  url (http://www.inf.fu-berlin.de/styles/institute-600x400.jpg)  
  no-repeat;  
}
```

Schrifteigenschaften

- font-family Name der Schriftfamilie
 - Generische
 - serif (Times)
 - sans-serif (Helvetica)
 - cursive (*Monotype Corsiva*) (Normale Schrift kursiv: *a*, Kursive Schrift: *a*)
 - fantasy (**Comic Sans**)
 - monospace (**Courier**, Lucida Console)
 - Konkrete Schriftnamen (Arial, Tahoma, Swiss, Garamond)
 - Auswahlversuch in Reihenfolge der Liste
`body { font-family: "new century school book", serif }`
- font-style Schriftstil
 - normal, *italic*, oblique
- font-variant Schriftvariante
 - normal, small-caps (KAPITÄLCHEN)

Schrifteigenschaften

- font-weight Stärke der Schrift (Schriftzug)
 - Absolute Werte: normal, bold, 100, 200, 300, 400, 500, 600, 700, 800, 900,
 - Relative Werte: bolder, lighter
- font-size Größe der Schrift
 - Absolute Werte: xx-small, x-small, small, medium, large, x-large, xx-large, Größenangabe
 - Relative Werte: larger, smaller
- font *Stil Variante [Stärke] Größe Zeilenabstand*
Schriftfamilie
 - Kurznotation

Texteigenschaften

- word-spacing: *Länge*
Zusätzlicher Wortabstand
- letter-spacing: *Länge*
Zusätzlicher Buchstabenabstand (Dicke)
- white-space Legt Behandlung von Leerraum fest
 - normal: Leerzeichen fallen zusammen, automatischer Zeilenumbruch (<p>)
 - pre: Leerraum wird beachtet (<pre>...</pre>)
 - nowrap: Kein automatischer Zeilenumbruch mit

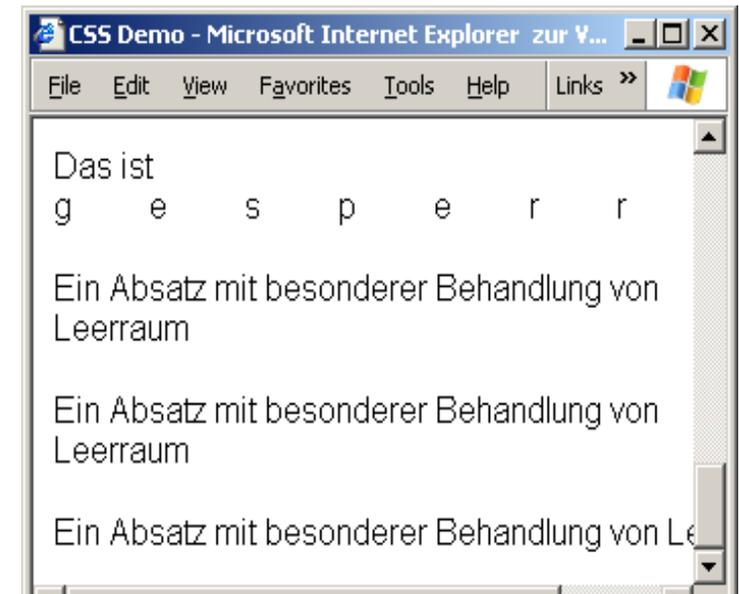
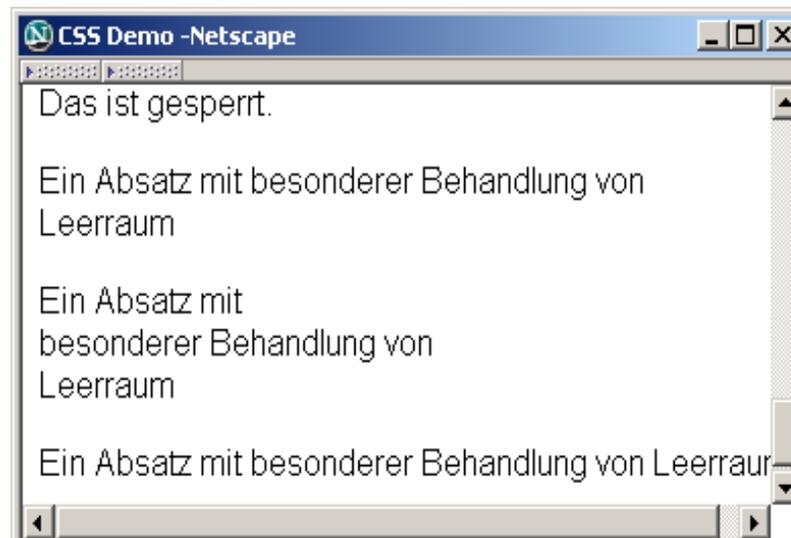
Demo

```
<p>Das ist <span style="letter-spacing:1cm">gesperrt</span>. </p>
```

```
<p style="white-space:normal">Ein Absatz mit besonderer Behandlung von Leerraum</p>
```

```
<p style="white-space:pre">Ein Absatz mit besonderer Behandlung von Leerraum</p>
```

```
<p style="white-space:nowrap">Ein Absatz mit besonderer Behandlung von Leerraum</p>
```



Texteigenschaften

- text-decoration Textdekoration
 - Werte: none, underline, overline, line-through, blink
- text-transform Textveränderung
 - Werte: capitalize, uppercase, lowercase, none
- vertical-align Ausrichtung eines Kastens

- baseline



- middle



- super



- sub



- text-top



- text-bottom



- top



- bottom



Texteigenschaften

- `text-align` Horizontale Ausrichtung
 - Werte: `left`, `right`, `center`, `justify`
- `text-indent`: *Länge* Einzug der ersten Zeile
- `line-height` Zeilenabstand
 - Faktor auf Schriftgröße
 - Länge
 - Prozent

- Cascading Style Sheets definieren Darstellung von Tags durch Belegen von CSS-Attributen
- Während ursprünglich für HTML entworfen auch für XML nutzbar
- Darstellung vom eigenen Element `<price>` weiss auf schwarz:

```
price {  
    color: white;  
    background-color: black;  
}
```
- CSS Attribute für visuelle oder auditive Ausgabe von Texten geeignet
- www.w3.org/1999/06/REC-xml-stylesheet-19990629

- Tabellen als Darstellungsobjekt
- Eigenschaften für Tabellen
- Diverse zusätzliche Eigenschaften für Text und Schriften
- ...

CSS2: Medienarten

- Darstellungsstil ist abhängig vom Ausgabemedium
 - Bildschirm
 - Papier
 - Spache
 - Braille
 - ...
- CSS erlaubt getrennte Stildefinitionen:

...

```
a:link {  
  color: #000099;  
  text-decoration: none ;  
}
```

```
@media print {
```

```
  a:link, a:visited, a:hover, a:active, a:focus {  
    text-decoration: none;  
    color: blue  
  }
```

```
}
```

Definierte Medienarten

- Medienarten:
 - al l
 - aural Sprachausgabe
 - brai l l e Taktile Ausgabe
 - embossed Braille Drucker
 - handhel d Klein, monochrom
 - pri nt Drucker
 - proj ecti on Beamer, Foliendruck
 - screen Bildschirm
 - tty Textterminals
 - tv Geringe Auflösung eingeschränkte Interaktion

Mediengruppen

- Darstellungseigenschaften müssen jeweils von Darstellern für bestimmte Mediengruppen beachtet werden

Media Types	Media Groups			
	continuous/paged	visual/aural/tactile	grid/bitmap	interactive/static
aural	continuous	aural	N/A	both
braille	continuous	tactile	grid	both
emboss	paged	tactile	grid	both
handheld	both	visual	both	both
print	paged	visual	bitmap	static
projection	paged	visual	bitmap	static
screen	continuous	visual	bitmap	both
tty	continuous	visual	grid	both
tv	both	visual, aural	bitmap	both

- Zusätzliche Eigenschaften
 - Seite ist umgebender Darstellungskasten
 - page- Eigenschaften
 - Es treten Seitenumbrüche auf
 - Eigenschaften page-break-before, page-break-after, page-break-inside
 - Werte: auto, always, avoid, left, right
 - Eigenschaften orphans, widows Hurenkinder/Schusterjungen

Schusterjunge / widow



Hurenkind / orphan



Darstellung gesprochener Sprache

- Darstellungseigenschaften betreffen
 - Ton
 - Dauer und Reihenfolge
 - Spracheigenschaften

- Beispiel

```
H1, H2, H3, H4, H5, H6 {  
    voi ce-fami l y: paul ;  
    stress: 20;  
    ri chness: 90;  
    cue-before: url ("pi ng. au")  
}  
P. hei di { azi muth: center-l eft }  
P. peter { azi muth: ri ght }  
P. goat { vol ume: x-soft }
```

Zusammenfassung

- CSS trennt Inhalt und Darstellung
- CSS ist getrennte Sprache für Darstellungseigenschaften
- Vielfältige Einstellungsmöglichkeiten
- CSS2: Erweiterung auf andere Medienarten